SOFT MAIL

การทำข้อสอบ ปรนัย 4 ตัวเลือก ตอนที่ 2 สร้างแบบทดสอบ

แสดงให้คุณเห็นถึงวิธีการเขียนโปรแกรมเพื่อใช้ในการสร้างแบบทดสอบ

ในบทความตอนที่ 1 นั้นเราได้เรียนรู้เรื่องการสร้างเครื่องคิดเลขใช้ประกอบการสอบและนำไปใช้งานอื่น ๆ เช่น ใช้ร่วมกับโปรแกรมพิมพ์เอกสารราชวิถีเวิร์ดพีซี เป็นต้น

สำหรับตอนที่สองนี้ เราจะมาเขียนโปรแกรมสร้างแบบทดสอบกันต่อไป

โดยผู้เขียนกำหนดให้ข้อสอบแต่ละข้อที่สร้างนั้นสามารถแสดงข้อความเส้นตรง รูปสี่เหลี่ยม วงกลม หรือวงรีได้ ซึ่งจะทำให้สามารถนำไปเขียนข้อสอบได้ทุกรายวิชา รวมทั้งเก็บคำตอบของแต่ละข้อไว้ในไฟล์เดียวกันด้วย ไฟล์หนึ่งไฟล์จะใช้ในการเก็บข้อสอบได้หนึ่งข้อ หากต้องการข้อสอบหลายข้อก็ต้องเก็บลงในไฟล์ที่มีชื่อไม่ซ้ำกัน

ซอฟต์เมล์ของตอนที่ 2 นี้ประกอบด้วยไฟล์ 6 ไฟล์คือ thai.tpu, head.tpu, calculat.tpu, headtest.pas, editgr.pas และ edittest.pas สำหรับไฟล์ *.tpu นั้นคอมพายล์ด้วยเทอร์โบปาสคาล 5.5 ผู้อ่านที่ใช้เทอร์โบปาสคาลรุ่นอื่น ๆ คงต้องย้อนไปใช้บริการซอฟต์เมล์ฉบับก่อนหน้านี้ด้วย

เนื่องจากไฟล์ editgr.pas นั้นมีความยาวถึง 488 บรรทัด ผู้เขียนจึงขอแจ้งให้ทราบล่วงหน้าตรงนี้ว่า ในบทความไม่มีลิสติงของไฟล์นี้มาให้ แต่ได้แจ้งรายละเอียดของการทำงานของชุดคำสั่งต่าง ๆ ให้ทราบเพื่อผู้อ่านจะได้ทดลองความรู้โดยเขียนขึ้นมาเอง ท่านที่อยากเห็นลิสติงคงต้องใช้บริการซอฟต์เมล์ครับ

ปรับปรุงยูนิต CALCULAT.TPU

เนื่องจากไฟล์ calculat.tpu ในตอนที่ 1 นั้นผู้เขียนมุ่งให้ใช้งานได้เอกเทศ จึงได้กำหนดให้ออกจากโปรแกรมแล้วใช้คำสั่ง closegraph; halt; ซึ่งมีผลให้ออกจากกราฟิกและไปยังดอสซึ่งเป็นการสิ้นสุดโปรแกรมทั้งหมด เมื่อนำมาใช้ในตอนที่ 2 นี้จึงต้องปรับปรุงส่วนนี้ เพราะว่าเราจะใช้เครื่องคิดเลขช่วยในการคำนวณหา

คำตอบจากข้อสอบที่เป็นโจทย์คำนวณ ซึ่งเมื่อออกจากเครื่องคิดเลขจะต้องกลับมาที่ส่วนการเขียนข้อสอบ ไม่ใช่จบการทำงานและออกไปดอส

ส่วนที่ปรับปรุงมีทั้งสิ้น 6 รายการ ดังนี้

1. เปลี่ยนชื่อค่าคงที่ในบรรทัดที่ 6 (รูปที่ 2) จาก key เป็น keymain

2. เพิ่มตัวแปรต่อจากบรรทัดที่ 22 (รูปที่ 2) อีกหนึ่งตัวแปรคือ

```
key:array[1..6],1..4] of char;
```

3. บรรทัดที่ 157 (รูปที่ 6) ตัดของเดิมออกทั้งหมดแล้วป้อน ch = #255; Error:=true; แทน

4. บรรทัดที่ 257 (รูปที่ 7) เพิ่ม #255 เป็น

```
until ch in ['-', '+', '*', '=', '/', #255];
```

5. เพิ่มคำสั่งต่อจากคำสั่ง begin ในบรรทัดที่ 338 (รูปที่ 10) ดังนี้

```
for Ycal:=1 to 6 do
  for Xcal:=1 to 4 do
    key[Ycal][Xcal]:=keymain[Ycal][Xcal];
```

6. ตัดบรรทัดที่ 390-401 (รูปที่ 10) ออกไป

เมื่อคอมพายล์จะได้ไฟล์ calculat.tpu ที่นำไปใช้งานในการสร้างแบบทดสอบ และใช้ในบทความตอนที่ 3 ด้วย

สร้างยูนิตเริ่มต้น

ยูนิตไฟล์ที่เรานำมาใช้งานร่วมกับโปรแกรมสร้างแบบทดสอบมีดังนี้

1. ไฟล์ thai.tpu เพื่อเข้าสู่กราฟิกและแสดงข้อความภาษาไทย

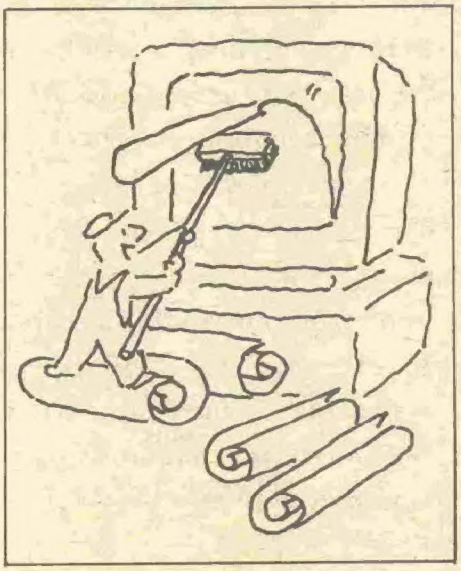
2. ไฟล์ head.tpu ใช้ในการตรวจสอบเกี่ยวกับเม้าส์ การแสดงกระดาน 3 มิติ

```
 1: UNIT HEADTEST;
 2: INTERFACE
 3: uses crt,dos,graph,thai,head;
 4: type  Test_Rec=record
 5:                 Code:char;                {1}
 6:                 Val:array[1..4] of word;  {8}
 7:                 Patt,Cpatt,Cline,Bold:byte; {4}
 8:                 Width,StAng,EndAng:integer; {6}
 9:                 Text:string[100];         {101}
10:               end;                  {total=120}
11: var Xasp,Yasp,Xline,Yline,Xwidth,Ywidth,Xright,
12:     XlP,YuP,XrP,YdP,Rx,Ry,Cx,Cy:word;
13:     ST,STT,Filename:string;  ck:integer;
14:     Pic,Ppic:pointer;  SaveFile:boolean;
15:     TestRec:Test_Rec;  NoRec:byte;
16:     TestMain:array[0..200] of Test_Rec;
17:     Poly:array[1..72] of pointtype;
18:     KbdFlag:byte absolute $0040:$0017;
19: PROCEDURE SoundOn(f:word);
20: PROCEDURE ShowLine;
21: PROCEDURE ShowBox;
22: PROCEDURE CircleShow(Mode:char);
23: PROCEDURE Display(NoST,NoEnd:word; mode:byte);
24: FUNCTION  ReadSTR(x,y:word; Count:byte; mode:char):string;
25: PROCEDURE ReadTest;
26: IMPLEMENTATION
```

รูปที่ 1
ส่วนกำหนดตัวแปรและชุดคำสั่งในยูนิต headtest.tpu

3. ไฟล์ calculat.tpu เรียกใช้เครื่องคิดเลขมาช่วยในการคำนวณโจทย์

ส่วนที่เราจะสร้างเพิ่มอีกหนึ่งยูนิตคือ headtest.tpu ซึ่งจะนำไปใช้ในบทความตอนที่ 3 ด้วย โดยจะมีชุดคำสั่งในการลากเส้นตรง วาดรูปสี่เหลี่ยม วาดวงกลมหรือวงรี และแสดงข้อความ โดยเป็นการนำเอาข้อมูลที่อ่านได้จากไฟล์ที่เก็บแบบทดสอบมาแสดงตามเงื่อนไขที่เราได้สร้างไว้



เรามาดูส่วนหัวของยูนิตไฟล์ที่เป็นส่วนหนึ่งของไฟล์ headtest.pas ดังรูปที่ 1 มีรายละเอียดดังนี้

1. บรรทัดที่ 4-10 กำหนดโครงสร้างของระเบียนข้อมูลของไฟล์ข้อสอบ ซึ่งมีความหมายดังนี้

code รูปแบบการแสดงผล (W ข้อความ L เส้นตรง B รูปสี่เหลี่ยม C วงกลมหรือวงรี)

Val[1]–Val[4] เก็บจุดปลายของเส้นตรง หรือมุมบนซ้ายและมุมล่างขวาของรูปสี่เหลี่ยม หรือจุดศูนย์กลางของวงกลม รัศมีแกนนอนและรัศมีแกนตั้ง

Patt,Cpatt,Cline,Bold เก็บรูปแบบพื้น สีพื้น สีของเส้น และความหนา/บางของเส้น

Width,StAng.EndAng ความกว้าง มุมเริ่มต้น มุมสิ้นสุดของวงกลมหรือวงรี

2. บรรทัดที่ 11-18 ตัวแปรที่เกี่ยวข้อง โดยตัวแปร TestMain นั้นสำหรับเก็บข้อมูลต่าง ๆ ของแบบทดสอบที่เราสร้างขึ้น โดยผู้เขียนใช้ testmain

[0].code เก็บคำตอบถูกของข้อสอบนั้นไว้ พร้อมเก็บจำนวนระเบียนข้อมูลทั้งหมดใน testmain[0].Val[1] ตัวแปรทั้งหมดที่กำหนดไว้นี้สามารถอ้างถึงในโปรแกรมที่เราเรียกใช้ยูนิตนี้ ส่วนตัวแปร testrec จะใช้สำรองตัวแปร testmain[ ] เพื่อนำมาแสดงผลบนจอภาพ

3. บรรทัดที่ 19-25 ชุดคำสั่งที่เรียกใช้ได้ภายนอกยูนิต

```
27: PROCEDURE SoundOn(f:word);
28: begin
29:  sound(f); delay(1000); nosound;
30: end;

31: PROCEDURE ShowLine;
32: begin
33:   mouseoff;
34:   with TestRec do
35:    begin
36:     setlinestyle(0,255,Bold); setcolor(Cline);
37:     Line(Val[1],Val[2],Val[3],Val[4]);
38:    end;
39: end;

40: PROCEDURE ShowBox;
41: begin
42:   Mouseoff;
43:   with TestRec do
44:    begin
45:     setlinestyle(0,$FFFF,Bold);
46:     setfillstyle(Patt,Cpatt);      Setcolor(Cline);
47:     if Patt>0 then bar(Val[1],Val[2],Val[3],Val[4]);
48:     Rectangle(Val[1],Val[2],Val[3],Val[4]);
49:     Xwidth:=Val[3]-Val[1];  Ywidth:=Val[4]-Val[2];
50:    end;
51: end;
```

รูปที่ 2
ชุดคำสั่งลากเส้นตรงและวาดรูปสี่เหลี่ยมจากข้อมูลที่กำหนด

นำข้อมูลมาลากเส้นตรงและวาดรูปสี่เหลี่ยม

เมื่อเราได้กำหนดโครงสร้างของไฟล์ข้อมูลแล้ว เรื่องต่อไปคือการออกแบบให้ข้อมูลเหล่านั้นสามารถนำมาแสดงภาพกราฟิก เช่น ข้อความ เส้นตรง รูปสี่เหลี่ยม และวงกลมได้ ดังนั้นเราจึงต้องเริ่มต้นในการเขียนชุดคำสั่งเพื่อการนี้ก่อน

ในการลากเส้นตรงและวาดรูปสี่เหลี่ยมนั้นค่อนข้างง่าย เพราะเราสามารถใช้คำสั่งที่ให้มาในเทอร์โบปาสคาลได้เลย ซึ่งสามารถนำมาเขียนโปรแกรมได้ดังรูปที่ 2 มีรายละเอียดดังนี้

1. บรรทัดที่ 27-30 ชุดคำสั่งในการเปล่งเสียง โดยสามารถกำหนดความถี่ของเสียงได้

2. บรรทัดที่ 31-39 ชุดคำสั่งในการลากเส้นตรง

3. บรรทัดที่ 33 ก่อนที่จะวาดรูปกราฟิกใด ๆ นั้น เราจะต้องสั่งให้คืนภาพเดิมให้กับจอภาพก่อน โดยใช้คำสั่ง mouseoff

4. บรรทัดที่ 34 การอ้างถึงเขตข้อมูลที่อยู่ในระเบียนข้อมูลนั้น โดยอ้างถึง "ระเบียนข้อมูล" ในชั้นแรกก่อน

5. บรรทัดที่ 36 กำหนดความหนา/บางของเส้น และสีของเส้น ซึ่งจะเห็นว่าเราอ้างถึงเขตข้อมูลได้เลย หากเราไม่มีบรรทัดที่ 34 ต้องเขียนการกำหนดสีเส้นตรงเป็น setcolor(testrec.Cline) หรือการกำหนดความหนา/บางต้องเขียนเป็น setlinestyle(0,255, testrec.Bold)

6. บรรทัดที่ 37 ลากเส้นตรงด้วยตัวแปร Val[1] - Val[4] ซึ่งถ้าไม่มีบรรทัดที่ 34 เราจะต้องอ้างถึงเขตข้อมูลโดยตรง คือ testrec.val[1]–testrec.val[4]

7. บรรทัดที่ 40-51 ชุดคำสั่งในการวาดรูปสี่เหลี่ยม

8. บรรทัดที่ 42 ใช้คำสั่ง mouseoff เพื่อคืนภาพเดิมให้กับจอภาพ

9. บรรทัดที่ 43 อ้างถึงระเบียนข้อมูล testrec

10. บรรทัดที่ 45 กำหนดความหนา/บางของเส้นขอบ

11. บรรทัดที่ 46 กำหนดรูปแบบสีของพื้น และสีของเส้นขอบ

12. บรรทัดที่ 47 ถ้ารูปแบบพื้นไม่ใส่หมายเลข 0 ให้วาดรูปสี่เหลี่ยมทึบด้วยคำสั่ง Bar

13. บรรทัดที่ 48 วาดรูปสี่เหลี่ยม

14. บรรทัดที่ 49 หาความกว้างและความสูงของรูปสี่เหลี่ยม (ซึ่งจะนำไปใช้งานในตอนต่อไป)

สร้างวงกลมหรือวงรีใหม่

ความจริงแล้วคำสั่งเกี่ยวกับวงกลมหรือวงรีในเทอร์โบปาสคาลนั้นมีมาให้หลายคำสั่ง แต่ผลปรากฏว่าไม่สามารถสนองตอบต่อความต้องการบางอย่างได้ เช่น ไม่สามารถเขียนวงกลมในลักษณะของวงแหวนได้ หากใช้วิธีเขียนวงกลมทึบ 2 วงซ้อนกัน โดยวงในกำหนดสีพื้นเป็นสีดำ (ซึ่งจะได้วงแหวน) แต่ภาพเดิมที่อยู่ในวงแหวนจะถูกเขียนทับด้วยวงกลมทึบวงในด้วย ซึ่งไม่ใช่การวาดรูปวงแหวนที่ถูกต้อง

ผู้เขียนจึงจำเป็นต้องสร้างชุดคำสั่งขึ้นมาใหม่เพื่อให้สามารถทำงานได้มากขึ้น โดยหนึ่งคำสั่งนี้จะวาดรูปวงกลมหรือวงรีได้หลายรูปแบบขึ้นอยู่กับการกำหนดค่าของตัวแปรที่เกี่ยวข้อง

```
52: PROCEDURE NewCircle(x,y:integer; StAng,EndAng,
53:                         XRadius,YRadius,Width:word);
54: var k:word; sector:boolean;
55:     xx,yy,Angle:integer;
56: begin
57:  sector:=true;
58:  if (Width>=Xradius) or (Width>=Yradius) then Width:=0;
59:  if ((StAng=0) and (EndAng=360)) or (Width<>0) then sector:=false;
60:  k:=1;
61:  if sector then
62:   begin
63:     Poly[1].x:=x; Poly[1].y:=y;  k:=2;
64:   end;
65:   angle:=StAng;
66:   repeat
67:     xx:=round(XRadius*cos((Angle/180)*pi));
68:     yy:=round(YRadius*sin((Angle/180)*pi)*(Xasp/Yasp));
69:     Poly[k].x:=x+xx; Poly[k].y:=y-yy;
70:     Inc(Angle,4);  inc(k);
71:   until Angle>=EndAng;
72:    xx:=round(XRadius*cos((EndAng/180)*pi));
73:    yy:=round(YRadius*sin((EndAng/180)*pi)*(Xasp/Yasp));
74:    Poly[k].x:=x+xx; Poly[k].y:=y-yy;  inc(k);
75:  if width<>0 then
76:   begin
77:    angle:=EndAng;
78:    repeat
79:     xx:=round((XRadius-Width)*cos((Angle/180)*pi));
80:     yy:=round((YRadius-Width)*sin((Angle/180)*pi)*(Xasp/Yasp));
81:     Poly[k].x:=x+xx; Poly[k].y:=y-yy;
82:     dec(Angle,4);  inc(k);
83:    until Angle<=StAng;
84:     xx:=round((XRadius-Width)*cos((StAng/180)*pi));
85:     yy:=round((YRadius-width)*sin((StAng/180)*pi)*(Xasp/Yasp));
86:     Poly[k].x:=x+xx; Poly[k].y:=y-yy;
87:     inc(k); poly[k]:=poly[1];
88:   end
89:   else
90:      Poly[k]:=poly[1];
91:   if Width=1 then drawPoly(k,Poly) else  fillPoly(k,Poly);
92: end;

93: PROCEDURE CircleShow(Mode:char);
94: begin
95:   setviewport(135,0,getmaxx,398,clipon);
96:   setfillstyle(TestRec.patt,TestRec.Cpatt);
97:   setcolor(TestRec.Cline);
98:   setlinestyle(0,$FF,TestRec.Bold);
99:   with TestRec do
100:   begin
101:    if TestRec.patt=0 then
102:       NewCircle(val[1]-135,Val[2],StAng,EndAng,Val[3],Val[4],1)
103:     else
104:      NewCircle(val[1]-135,Val[2],StAng,EndAng,Val[3],Val[4],width);
105:     if mode='O' then
106:       NewCircle(val[1]-135,Val[2],0,360,Val[3],Val[4],1);
107:   end;
108:  setlinestyle(0,$FF,1);
109:  setviewport(0,0,getmaxx,getmaxy,clipon);
110: end;

111: PROCEDURE Display(NoST,NoEnd:word; mode:byte);
112: var No:byte;
113: begin
114:  if mode=0 then clsgr(135,0,getmaxx,390);
115:  for no:=NoST to NoEnd do
116:   begin
117:    TestRec:=TestMain[No]; Mouseoff;
118:    case TestRec.code of
119:     'B':ShowBox;
120:     'L':ShowLine;
121:     'C':CircleShow('C');
122:     'W':with TestRec do
123:          begin
124:           setcolor(TestRec.Cline);
125:           OutThaiXy(Val[1],Val[2],#17+Text);
126:          end;
127:    end;
128:   end;
129:   Mouseon;
130: end;
```

รูปที่ 3
ชุดคำสั่งวาดวงกลมหรือวงรี และการแสดงข้อสอบบนจอภาพ

โดยชุดคำสั่งในการสร้างวงกลมใหม่จะเป็นดังรูปที่ 3 มีรายละเอียดดังนี้

1. บรรทัดที่ 57 กำหนดการวาดวงกลมหรือวงรีแบบไม่เต็มวง

2. บรรทัดที่ 58-59 ตรวจสอบค่าที่กำหนดมาให้ว่าเป็นการวาดรูปทึบหรือวาดเต็มวง

3. บรรทัดที่ 61-64 ถ้าเป็นการวาดเต็มวง เราจะกำหนดให้ค่าเริ่มต้นของตัวแปร poly อยู่ที่จุดศูนย์กลางของวงกลมหรือวงรีนั้น

4. บรรทัดที่ 65 กำหนดมุมเริ่มต้นด้วยค่ามุมแรกที่ให้มาเมื่อเรียกใช้ชุดคำสั่งนี้

5. บรรทัดที่ 66-71 วนรอบเพื่อคำนวณค่าให้กับตัวแปร poly ด้วยรัศมีที่กำหนดมาให้ จนกว่าค่าของมุมจะมาถึงมุมสุดท้ายที่กำหนดให้

6. บรรทัดที่ 72-74 คำนวณหาค่าให้ตัวแปร poly ณ มุมสุดท้ายของการวาดรูป

7. บรรทัดที่ 75-88 ในกรณีที่ค่า width ไม่เท่ากับ 0 (กำหนดให้วาดรูปวงแหวน) จะทำการคำนวณหาค่า poly ในรัศมีที่เล็กลงเท่าความกว้างที่กำหนดให้ โดยวนรอบย้อนจากมุมโตไปหามุมเล็กสุด

8. บรรทัดที่ 90 ในกรณีที่วาดวงทึบเต็มวงจะกำหนดให้ตัวแปร poly ค่าสุดท้ายเท่ากับค่าเริ่มต้น

9. บรรทัดที่ 91 วาดวงกลมโปร่งหรือวาดวงกลมทึบตามความกว้างที่กำหนดมา

10. บรรทัดที่ 93-110 ชุดคำสั่งแสดงวงกลมหรือวงรี โดยใช้ข้อมูลจากโครงสร้างไฟล์

11. บรรทัดที่ 95 กำหนดขอบเขตของจอภาพให้อยู่เฉพาะส่วนที่ว่างของจอภาพเท่านั้น โดยกำหนดให้เป็

clipon เพื่อป้องกันการวาดวงกลมหรือวงรีล้ำเข้าไปในส่วนอื่น ๆ ที่ใช้เป็นคำอธิบาย

12. บรรทัดที่ 96-98 กำหนดรูปแบบและสีของพื้นวงกลม สีของเส้นรอบวง และความหนา/บางของเส้นรอบวง

13. บรรทัดที่ 98-107 ส่วนการวาดรูปวงกลมหรือวงรีตามข้อมูลที่กำหนดไว้

14. บรรทัดที่ 108-109 กำหนดขนาดของเส้นให้เป็นปกติ และกำหนดขอบเขตของกราฟิกให้เต็มจอตามปกติ

15. บรรทัดที่ 111-130 ส่วนการแสดงผลตามค่าตัวแปรที่เก็บไว้ในระเบียนข้อมูล

16. บรรทัดที่ 117 กำหนดให้ตัวแปร testrec เท่ากับตัวแปร test-main[no] เพื่อสะดวกในการนำไปแสดงผล

17. บรรทัดที่ 119-121 วาดรูปสี่เหลี่ยม ลากเส้นตรง และวาดวงกลมหรือวงรี

18. บรรทัดที่ 122-126 แสดงข้อความด้วยสีที่กำหนดไว้แล้วบนจอภาพ

19. บรรทัดที่ 129 เมื่อสิ้นสุดการทำงานด้านกราฟิก ก็จะกลับมาแสดงเคอร์เซอร์ของเม้าส์ใหม่

ชุดคำสั่งป้อนข้อความ

ในการเขียนข้อสอบนั้นเราจะต้องมีการป้อนข้อความเป็นหลัก แต่เนื่องจากโปรแกรมสร้างแบบทดสอบนี้เราจะมีการวาดรูปกราฟิกประกอบด้วย ดังนั้นการป้อนข้อความจึงแตกต่างจากการป้อนข้อความบนพื้นสีดำ ซึ่งไม่ต้องคำนึงว่าจะมีภาพใด ๆ อยู่ในบริเวณการป้อนข้อความหรือไม่ การป้อนข้อความในโปรแกรมนี้ต้องคงภาพที่อยู่ในบริเวณนั้น ๆ ไว้ได้ โดยต้องป้อนข้อความซ้อนทับลงไปในภาพเดิมเท่านั้น

หลักการที่ผู้เขียนใช้ในการป้อนข้อความแบบนี้มีดังนี้

1. ก่อนการป้อนข้อความใด ๆ จะจำภาพหน้าจอในกรอบสี่เหลี่ยมจากตำแหน่งเริ่มต้นของข้อความไปจนสิ้นสุดจอภาพด้านขวา โดยมีคำสั่งคำนวณหาขอบเขตของกรอบสี่เหลี่ยมนี้ในโปรแกรม

2. จำภาพหน้าจอบริเวณที่จะแสดงเคอร์เซอร์ไว้ในตัวแปรอีกตัวหนึ่ง ทั้งนี้เพื่อให้การแสดงเคอร์เซอร์ของการป้อนข้อความยังคงแสดงภาพเดิมสลับกับเคอร์เซอร์รูปสี่เหลี่ยม

3. เคอร์เซอร์รูปสี่เหลี่ยมในกรณีที่อยู่ในโหมดการป้อนภาษาไทย/ภาษาอังกฤษจะมีขนาดใหญ่/เล็กตามลำดับ

4. การสลับภาษาไทย/ภาษาอังกฤษ จะใช้การกดแป้น Esc

5. เมื่อกดแป้นพิมพ์เพื่อป้อนข้อความ จะเริ่มต้นการคืนภาพบริเวณเคอร์เซอร์แสดงก่อน ถ้าตรวจพบว่าอักขระนั้นเป็นอักขระข้อความก็จะคืนภาพหน้าจอ (ตามข้อ 1) ตามด้วยแสดงข้อความที่ได้นำอักขระนั้น ๆ ไปต่อท้ายไว้แล้ว

6. สามารถกำหนดการป้อนเฉพาะตัวเลข หรือกำหนดให้ป้อนชื่อไฟล์ได้ (มีอักขระบางตัวที่ไม่ยอมให้ใช้ในการตั้งชื่อไฟล์)

ส่วนการทำงานอื่น ๆ ก็จะคล้ายกับการป้อนข้อความในกราฟิกทั่ว ๆ ไป ซึ่งเราสามารถเขียนชุดคำสั่งนี้ได้ดังรูปที่ 4 มีรายละเอียดดังนี้

1. บรรทัดที่ 133-139 ค่าคงที่ที่ใช้เปลี่ยนอักขระภาษาอังกฤษที่ได้จากการกดแป้นพิมพ์ให้เป็นภาษาไทยเมื่อตรวจพบว่าในขณะนั้นอยู่ในโหมดการป้อนภาษาไทย

2. บรรทัดที่ 144 กำหนดค่าตัวแปรที่เกี่ยวข้อง โปรดสังเกตว่าก่อนการป้อนข้อความเราจะต้องสั่งให้เม้าส์คืนภาพแก่จอภาพก่อนเสมอ

3. บรรทัดที่ 145-152 ตรวจสอบว่าอยู่ในการป้อนข้อความแบบใด

4. บรรทัดที่ 153 จำภาพหน้าจอไว้ด้วยตัวแปร Ppic

5. บรรทัดที่ 155 เก็บภาพหน้าจอส่วนที่จะแสดงเคอร์เซอร์ด้วยตัวแปร Pcursor

6. บรรทัดที่ 156-167 ส่วนการแสดงเคอร์เซอร์ในการป้อนข้อความ

7. บรรทัดที่ 168 เมื่อมีการกดแป้นให้คืนภาพ Pcursor

8. บรรทัดที่ 170-172 ตรวจการกดแป้นในกรณีที่กำหนดให้ป้อนชื่อไฟล์

9. บรรทัดที่ 173-174 ตรวจการกดแป้นในกรณีที่กำหนดให้ป้อนตัวเลข

10. บรรทัดที่ 175-176 เปลี่ยนส่วนการป้อนให้เป็นภาษาไทยในกรณีที่อยู่ในโหมดป้อนภาษาไทย

11. บรรทัดที่ 178-181 การเปลี่ยนโหมดการป้อนภาษาไทย/ภาษาอังกฤษด้วยแป้น Esc

12. บรรทัดที่ 182-189 ยอมรับการป้อนข้อความเมื่ออักขระที่ป้อนอยู่ในขอบเขตที่กำหนด ซึ่งจะต้องคืนภาพ Ppic ให้จอภาพก่อนการแสดงข้อความ

13. บรรทัดที่ 186-187 กรณีที่ไม่ใช่สระบน สระล่าง หรือวรรณยุกต์จะเลื่อนเคอร์เซอร์ไปทางขวา

14. บรรทัดที่ 190-197 การลบข้อความด้วยการกดแป้นลูกศรถอยหลัง

15. บรรทัดที่ 200 กำหนดให้ตัวแปรที่เรียกใช้ฟังก์ชันนี้เท่ากับข้อความที่ป้อนนี้

16. บรรทัดที่ 202-214 ชุดคำสั่งในการอ่านข้อสอบจากแผ่นดิสค์มาแสดง

```
131: FUNCTION ReadSTR(x,y:word; Count:byte; mode:char):string;
132: const
133:  ThaiChr:array[33..123] of integer=(46,46,50,51,52,55,165,54,
134:     55,53,57,191,162,211,187,166,195,47,45,190,180,215,219,163,
135:     179,169,196,176,168,201,165,49,194,37,167,173,172,210,170,
136:     223,177,227,198,197,63,228,214,171,48,175,164,182,226,203,
137:     34,41,222,40,184,92,195,216,56,96,189,217,209,161,207,178,
138:     208,225,193,224,206,199,181,220,183,192,213,188,200,204,
139:     218,202,212,185,221,186,174);
140: var No,k:byte; Xold:word; High:integer;
141:     Thaimode,NumMode,FileMode:boolean;
142:     Pcursor:array[1..3000] of byte;
143: begin
144:   ST:=''; k:=0; Xold:=x; no:=0; mouseoff; inc(y,8); high:=-8;
145:   ThaiMode:=false; filemode:=false; numMode:=false;
146:   case Mode of
147:    'T':begin
148:          ThaiMode:=true; High:=2;
149:        end;
150:    'N':Nummode:=true;
151:    'F':filemode:=true;
152:   end;
153:   getimage(x,y-10,getmaxx,y+15,ppic^);
154:   repeat
155:     getimage(x,y-8,x+10,y+14,PCursor);
156:    repeat
157:      regs.AH:=$2C; intr($21,regs);
158:      if (regs.DL in [0..49]) and (k=0) then
159:       begin
160:         setfillstyle(1,TestRec.Cline);
161:         bar(x,y-High,x+8,y+9); k:=27;
162:       end;
163:      if (regs.DL in [50..99]) and (k=27) then
164:       begin
165:         putimage(x,y-8,PCursor,0); k:=0;
166:       end;
167:    until keypressed;
168:    readchar; putimage(x,y-8,PCursor,0); k:=0;
169:    if Filemode then
170:       if ch in [':','\',',',#32,'@','*','+',';','[',']',';',
171:             #27,'"','<','>','?','/','.'] then
172:                   soundOn(300) else ch:=upcase(ch);
173:    if nummode and (ch in [#27,#32..#47,
174:                   #58..#225]) then SoundOn(300);
175:    if ThaiMode and (ch in [#33..#123]) then
176:                   ch:=chr(Thaichr[ord(ch)]);
177:     case ch of
178:      #27:if ThaiMode then
179:          begin ThaiMode:=false; High:=-10; end
180:            Else
181:          begin ThaiMode:=true; High:=2; end;
182:     #32..#228:if no<Count then
183:                 begin
184:                   putimage(xold,y-10,ppic^,0);
185:                   ST:=ST+ch; outthaixy(Xold,Y-8,#17+St);
186:                   if ch<#215 then
187:                     begin inc(x,9); inc(no); end;
188:                   Xright:=x; if count=1 then ch:=#13;
189:                 end;
190:           #8:if No>0 then
191:                 begin
192:                   if ST[length(ST)]<#215 then
193:                     begin dec(no); dec(x,9); end;
194:                   ST:=copy(ST,1,length(ST)-1);
195:                   putimage(xold,y-10,ppic^,0);
196:                   outthaixy(Xold,Y-8,#17+ST);
197:                 end;
198:     end;
199:    until ch=#13;
200:    ReadSTR:=ST;
201: end;

202: PROCEDURE ReadTest;
203: var f:file; size:word;
204: begin
205:  assign(f,Filename);
206:  {$I-}  reset(f,1); {$I+}
207:  if IOResult=0 then
208:   begin
209:    size:=filesize(f);
210:    blockread(f,TestMain,size);  close(f);
211:    Norec:=TestMain[0].Val[1];
212:    Display(1,NoRec-1,0);  SaveFile:=false;
213:   end;
214: end;
215: {================ HEADTEST.PAS ================}
216: begin
217:  getmem(ppic,imagesize(0,0,getmaxx,30));
218: end.
```

รูปที่ 4 ชุดคำสั่งการป้อนข้อความและการอ่านเนื้อเรื่องของข้อสอบหนึ่งข้อ

ผลบนจอภาพ โดยได้จำนวนระเบียนข้อมูลเพื่อเขียนข้อสอบต่อด้วย ซึ่งวิธีนี้เราจะสามารถปรับปรุงข้อสอบเดิมในภายหลังได้

17. บรรทัดที่ 218 เมื่อเรียกใช้ยูนิตนี้จะจองเนื้อที่หน่วยความจำฮีพให้กับตัวแปร Ppic ทันที

เมื่อเราทำการคอมพายล์ไฟล์นี้จะได้ยูนิตไฟล์ชื่อ headtest.tpu ซึ่งในตอนที่ 3 ของบทความจะถูกเรียกใช้ด้วย

ยูนิต Editgr.tpu

สำหรับไฟล์ editgr.pas ซึ่งมีความยาวถึง 488 บรรทัดนั้น มีชุดคำสั่งสำคัญดังนี้

1. EditLine ชุดคำสั่งในการลากเส้นตรงด้วยเม้าส์ โดยสามารถเปลี่ยนจุดปลายได้ทั้งสองจุด รวมทั้งสามารถกำหนดสีและความหนา/บางของเส้นได้ด้วย

2. EditBox ชุดคำสั่งในการวาดรูปสี่เหลี่ยมด้วยเม้าส์ โดยสามารถเปลี่ยนมุมได้ทั้งสี่มุม เปลี่ยนรูปแบบของพื้น สีของพื้น สีของเส้นขอบ และความหนา/บางของเส้นขอบได้

3. SelectCircle ส่วนการเลือกว่าจะเขียนวงกลมเต็มวงหรือบางส่วนของวงเท่านั้น โดยถ้าเลือกการวาดวงกลมเพียงบางส่วน จะให้ป้อนมุมเริ่มต้นและมุมสิ้นสุด

4. EditCircle ชุดคำสั่งในการวาดวงกลมหรือวงรีด้วยเม้าส์ สามารถเคลื่อนย้ายจุดศูนย์กลางได้โดยง่าย เปลี่ยนรัศมีได้ทั้งแกนนอน (X) และแกนตั้ง (y) เปลี่ยนรูปแบบพื้นและสีของพื้น เปลี่ยนสีเส้นขอบและความหนา/บางของเส้นขอบได้

5. EraseOneRec ลบคำสั่งเดิมที่ไม่ต้องการได้ครั้งละหนึ่งคำสั่ง

ในแต่ละชุดคำสั่งจะมีข้อความที่ด้านล่างของจอภาพแจ้งให้ทราบว่าสามารถใช้แป้นพิมพ์แป้นใดบ้างในการเปลี่ยนแปลงรูปแบบของภาพตามชุด

คำสั่งที่ใช้งานอยู่ การเลิกงานแต่ละชุดคำสั่งจะต้องใช้เม้าส์เลือกไปยังกรอบสี่เหลี่ยมที่แสดงข้อความว่า "กลับเมนูหลัก" แล้วกดปุ่มของเม้าส์เท่านั้น ไม่สามารถย้อนกลับด้วยแป้นพิมพ์

สำหรับไฟล์ editgr.pas นี้ผู้เขียนจะนำลิสติงมาแสดงในบทความชุดต่อไป ซึ่งผู้เขียนตั้งใจว่าจะเป็นบทความเรื่องการทำข้อสอบสำหรับพิมพ์ลงกระดาษที่สามารถใช้เส้นตรง รูปสี่เหลี่ยม วงกลมหรือวงรี และแสดงข้อความได้อย่างสมบูรณ์ รวมทั้งมีสัญลักษณ์ทางคณิตศาสตร์และวิทยาศาสตร์อยู่ครบถ้วน โดยต้องสามารถพิมพ์ลงกระดาษด้วยเครื่องพิมพ์แบบ 24 หัวเข็มได้อย่างสวยงาม ซึ่งคงต้องติดตามในฉบับต่อ ๆ ไป

โปรแกรมสร้างแบบทดสอบ

เรามาศึกษาไฟล์หลักของการสร้างแบบทดสอบที่ชื่อ edittest.pas เป็นไฟล์สุดท้ายของบทความตอนที่ 2 นี้ ซึ่งมีส่วนสำคัญ 3 ส่วนคือ

1. การป้อนข้อความ ซึ่งการทำงานเริ่มด้วยการเลือกตำแหน่งเริ่มต้นของการป้อนข้อความด้วยเม้าส์ พร้อมกับเลือกสีของข้อความ (สีของเคอร์เซอร์) เมื่อได้ตำแหน่งตามต้องการก็จะเป็นการป้อนข้อความ

2. การเก็บข้อสอบลงแผ่นดิสก์ ซึ่งจะต้องให้คำตอบที่ถูกต้องของข้อสอบด้วย

3. ส่วนของโปรแกรมหลักที่มีเมนูให้เลือก การป้อนข้อความ ลากเส้นตรง วาดรูปสี่เหลี่ยม วาดวงกลมหรือวงรี เก็บข้อสอบ อ่านข้อสอบ ใช้เครื่องคิดเลข โดยทั้งหมดจะใช้เม้าส์เลื่อนเมื่อเลือกรายการ

```
 1: PROGRAM EDITTEST;
 2: uses crt,dos,graph,thai,head.calculat,Headtest,EditGR;
 3: PROCEDURE Entertext;
 4: var X,Y:word; k:byte;
 5:     Procedure Guide;
 6:      begin
 7:        Mouseoff;  Board3D(15,250,115,283,6,1);
 8:        OutThaixy(28,255,#17'กลับเมนูหลัก');
 9:        Board3D(15,287,115,320,11,1);
10:        OutThaixy(28,293,#17'ป้อนข้อความ');
11:        board3D(10,400,getmaxx,getmaxy,7,0);
12:        board3D(20,406,320,435,3,1);
13:        board3D(330,406,getmaxx-10,435,10,1);
14:        OutThaixy(40,410,#17'ส่วนการเลือกตำแหน่งเริ่มต้นของข้อความ');
15:        OutThaixy(340,410,#17'L => เปลี่ยนสีเคอร์เซอร์(สีของข้อความ)');
16:        OutThaixy(50,448,#17'เลือกตำแหน่งโดยการเลื่อนเคอร์เซอร์ของเมาส์'+
17:                                        'ไปยังจุดที่ต้องการ และกดปุ่มของเมาส์');
18:        TestRec.Val[1]:=x; TestRec.Val[2]:=y;
19:      end;
20:    Procedure InputText;
21:     begin
22:       mouseoff;  SaveFile:=true;
23:       setfillstyle(1,9); bar(12,245,117,322);
24:       board3D(10,400,getmaxx,getmaxy,7,0);
25:       board3D(20,406,getmaxx-10,435,3,1);
26:       OutThaixy(220,410,#17'ส่วนการป้อนข้อความ');
27:       OutThaixy(90,448,#17'ป้อนข้อความเสร็จให้กดแป้น Enter'+
28:                                ' เพื่อเลือกตำแหน่งการป้อนข้อความใหม่');
29:       setcolor(TestRec.Cline);  TestRec.Code:='W';
30:       TestRec.Text:=ReadSTR(TestRec.Val[1],TestRec.Val[2],90,'T');
31:       TestMain[NoRec]:=TestRec; inc(NoRec);  X:=Xright;
32:       TestMain[0].Val[1]:=Norec; getimage(x,y,x+10,y+10,ppic^);
33:       setposition(x,y);  Xmouse:=100; Guide;
34:     end;
35: begin
36:   Mouseoff; setfillstyle(1,9); bar(15,250,115,320);
37:   windowmouse(30,10,getmaxx-40,380); setposition(320,160);
38:   guide;  K:=0; X:=320; Y:=160;  ch:=#0;
39:   getimage(x,y,x+10,y+10,ppic^);
40:   repeat
41:       regs.AH:=$2C; intr($21,regs);
42:       if (regs.DL in [0..49]) and (k=0) then
43:        begin
44:          mouseoff; setfillstyle(1,TestRec.Cline);
45:          bar(x,y,x+8,y+10); k:=27;
46:        end;
47:       if (regs.DL in [50..99]) and (k=27) then
48:        begin
49:          mouseoff; k:=0; putimage(x,y,ppic^,0);
50:        end;
51:      MouseOn; regs.AX:=3; Intr($33,regs);
52:     if Regs.BX in [1,2] then
53:      begin
54:        MouseOff; findxy(Xmouse,Ymouse); ch:=#0; k:=0;
55:        putimage(x,y,ppic^,0);
56:        if (Xmouse>29) and (Xmouse<116) and
57:             (Ymouse>249) and (Ymouse<284) then Exit;
58:        if (Xmouse>29) and (Xmouse<116) and
59:           (Ymouse>286) and (Ymouse<321) then InputText;
60:        if Xmouse>135 then
61:          begin
62:            mouseoff; x:=Xmouse; y:=YMouse;
63:            TestRec.Val[1]:=x; TestRec.Val[2]:=y-8;
64:            getimage(x,y,x+10,y+10,ppic^)
65:          end;
66:      end;
67:      if keypressed then
68:       begin
69:        readchar;
70:        case upcase(ch) of
71:         'L':begin
72:               Inc(TestRec.Cline);
73:               if TestRec.Cline=16 then TestRec.Cline:=1;
74:             end;
75:        end;
76:        ch:=#0;
77:       end;
78:   until ch=#255;
79: end;
```

รูปที่ 5
ชุดคำสั่งในการเลือกตำแหน่งเริ่มต้นของข้อความและการป้อนข้อความ

เริ่มด้วยชุดคำสั่งในการป้อนข้อความในรูปที่ 5 กันเลย มีรายละเอียดของชุดคำสั่งดังนี้

1. บรรทัดที่ 2 ส่วนการเรียกใช้ยูนิตไฟล์ที่เกี่ยวข้อง

2. บรรทัดที่ 5-19 คำอธิบายซึ่งแสดงที่ด้านล่างของจอภาพให้ทราบว่ากำลังอยู่ในส่วนของการเลือกตำแหน่งของการป้อนข้อความ พร้อมกับแจ้งว่าเปลี่ยนสีข้อความได้ด้วยการกดแป้น L โดยบรรทัดที่ 18 จะเก็บตำแหน่งเริ่มต้นของข้อความไว้ใน val[1] และ val[2]

3. บรรทัดที่ 20-34 คำอธิบายเมื่ออยู่ในส่วนการป้อนข้อความ ซึ่งแสดงที่ด้านล่างของจอภาพเช่นกัน

4. บรรทัดที่ 29 กำหนดสีของข้อความด้วยตัวแปรสีของข้อความ และกำหนดให้ code = 'W' ซึ่งเป็นเงื่อนไขของการแสดงข้อความเมื่อเก็บลงไฟล์

5. บรรทัดที่ 30 ป้อนข้อความเก็บใน TestRec.Text

6. บรรทัดที่ 31-32 กำหนดค่าให้กับตัวแปรของการทำข้อสอบ

7. บรรทัดที่ 36-39 กำหนดขอบเขตของเม้าส์ ตำแหน่งของเม้าส์ และเก็บภาพหน้าจอ

8. บรรทัดที่ 40-50 แสดงเคอร์เซอร์ให้เลือกตำแหน่งเริ่มต้นของข้อความ

9. บรรทัดที่ 52-66 ตรวจสอบการทำงานเมื่อมีการกดปุ่มของเม้าส์

10. บรรทัดที่ 56-57 ถ้ากดปุ่มของเม้าส์ที่บริเวณข้อความว่า กลับเมนูหลัก จะออกจากชุดคำสั่ง

11. บรรทัดที่ 58-59 ถ้ากดปุ่มของเม้าส์บริเวณข้อความว่า ป้อนข้อความ จะเข้าสู่การป้อนข้อความ

12. บรรทัดที่ 67-77 ถ้ากดแป้น L จะเปลี่ยนสีของข้อความ (สีของเคอร์เซอร์จะเปลี่ยนด้วย)

13. บรรทัดที่ 78 ส่วนบังคับว่าชุดคำสั่งนี้ไม่สามารถออกได้ด้วยการกดแป้น

ในกรณีที่ต้องการแสดงข้อความยกกำลังหรือห้อยท้าย ให้แยกการป้อนเป็นคนละข้อความ โดยเลือกตำแหน่งของข้อความให้ตรงกับความต้องการด้วยตนเอง

เก็บข้อสอบลงแผ่นดิสค์

เมื่อเขียนข้อสอบที่มีทั้งข้อความ เส้นตรง รูปสี่เหลี่ยม วงกลมหรือวงรีได้จนพอใจ หรืออาจต้องการที่จะหยุดเป็นการชั่วคราวก่อน เราก็ต้องทำการเก็บข้อมูลของข้อสอบนั้น ๆ ลงไว้ในแผ่นดิสค์ก่อน โดยข้อสอบที่เก็บนี้จะต้องสามารถนำมาเขียนต่อหรือแก้ไขใน

```
80: PROCEDURE Showme;
81: begin
82:  mouseoff;  board3d(5,8,129,210,10,1);
83:  OutThaiXY(40,55,#17'โปรแกรม');
84:  OutThaiXY(30,90,#17'สร้างข้อสอบ');  mouseon;
85: end;

86: PROCEDURE SaveTest(Mode:byte);
87: var No,C:byte; f:file; Size:word;
88: begin
89:   Mouseoff; board3D(10,400,getmaxx,getmaxy,7,0);
90:   board3D(160,410,490,470,3,1);
91:   C:=TestRec.Cline; Testrec.Cline:=0;
92:   Outthaixy(200,430,#17#3'0ใส่ชื่อไฟล์ :');
93:    repeat
94:      FileName:=ReadSTR(300,430,8,'F');
95:    until filename<>'';
96:    No:=pos('.',Filename);
97:    if No<>0 then Filename:=copy(Filename,1,no)+'TST'
98:                  else Filename:=Filename+'.TST';
99:   setfillstyle(1,9);  bar(12,245,118,324);
100:  board3D(160,410,490,470,3,1);
101:  Outthaixy(200,430,#17#3'0ที่ต้องตอบถูก :');
102: repeat
103:  setfillstyle(1,3); bar(305,413,325,465);
104:  ST:=Readstr(310,430,1,'T'); Testmain[0].code:=upcase(ST[1]);
105: until Testmain[0].code in ['ก'..'ข','ค','ง','A'..'D','1'..'4'];
106: Testrec.Cline:=C;
107: assign(f,Filename);
108: {$I-} rewrite(f,1); {$I+}
109: if IOResult=0 then
110:  begin
111:   reset(f,1); size:=120*Norec; blockwrite(f,TestMain,size);
112:   close(f);   SaveFile:=false;
113:   if mode=1 then
114:    begin
115:     Outthaixy(55,430,#17#3'0โปรดเลือก');
116:     setfillstyle(1,7); bar(300,405,340,473);
117:     board3D(160,410,320,470,10,1);  board3D(330,410,490,470,6,1);
118:     Outthaixy(180,430,#17#3'0เขียนข้อสอบนี้ต่อ          เขียนข้อสอบข้อใหม่');
119:     windowmouse(163,412,488,468);  setposition(325,420); ch:=#0;
120:     repeat
121:      MouseOn; regs.AX:=3; Intr($33,regs);
122:       if Regs.BX in [1,2] then
123:        begin
124:          MouseOff; findxy(Xmouse,Ymouse);
125:          if (Xmouse>330) and (Xmouse<490) then
126:           begin
127:            clearMouse;  Norec:=1; ch:=#255;
128:            clsgr(135,0,getmaxx,390);  SaveFile:=true;
129:           end;
130:         if (Xmouse>160) and (Xmouse<320) then ch:=#255;
131:        end;
132:     until ch=#255;
133:    end;
134:  end;
135: end;
```

รูปที่ 6
ชุดคำสั่งในการเก็บข้อสอบลงแผ่นดิสค์

ภายหลังได้

ดังนั้นชุดคำสั่งต่อไปคือการเก็บข้อสอบลงแผ่นดิสก์ ซึ่งคือชุดคำสั่งในรูปที่ 6 มีรายละเอียดดังนี้

1. บรรทัดที่ 80-85 ส่วนแสดงให้ทราบว่าโปรแกรมที่กำลังทำงานนั้นเป็นโปรแกรมใด

2. บรรทัดที่ 92-95 ให้ป้อนชื่อไฟล์ โดยกำหนดให้ป้อนได้เพียง 8 ตัวเท่านั้น และจะต้องมีการป้อนชื่อไฟล์อย่างน้อย 1 ตัวถึงจะออกจากวนรอบ repeat ได้

3. บรรทัดที่ 96-98 ให้ชื่อสกุลของไฟล์ที่จะเก็บเป็น .TST

4. บรรทัดที่ 101-105 ให้คำตอบของข้อสอบนั้น ซึ่งจะสามารถเลือก ก-ง, A-D, 1-4 ก็ได้ (ขึ้นกับตัวเลือกที่ให้ไว้ในข้อสอบ

5. บรรทัดที่ 107-108 เปิดไฟล์ในชื่อที่ป้อน พร้อมกับตรวจสอบว่าการเปิดไฟล์นี้สมบูรณ์หรือไม่

6. บรรทัดที่ 109-134 ถ้าเปิดไฟล์ได้จะทำงานตามคำสั่งกลุ่มนี้

7. บรรทัดที่ 111-112 เก็บข้อมูลของข้อสอบทั้งหมดลงแผ่นดิสก์

8. บรรทัดที่ 113-133 ให้เลือกว่าจะเขียนข้อสอบนี้ต่อหรือจะเขียนข้อสอบข้อใหม่ ถ้าเขียนข้อสอบข้อใหม่จะทำงานตามคำสั่งบรรทัดที่ 125-129 ถ้าเลือกเขียนข้อเดิมต่อจะทำงานตามบรรทัดที่ 130

โปรแกรมหลัก

ตามแนวคิดที่เขียนมาทั้งหมดนี้ จะพบว่า เราได้ชุดคำสั่งในการทำงานมาครบถ้วนแล้ว โปรแกรมหลักจึงเพียงเรียกใช้ชุดคำสั่งเหล่านี้มาทำงานด้วยเท่านั้น ผู้เขียนได้กำหนดส่วนของโปรแกรมหลักไว้ดังนี้

```
136: PROCEDURE Main;
137: var Mode:char; p:byte;
138:       ProceDure Guide;
139:        begin
140:         Mouseoff; board3D(10,400,getmaxx,getmaxy,7,0);
141:         board3D(20,405,390,478,12,0);
142:         board3D(400,420,getmaxx-10,460,3,1);
143:         board3D(30,409,140,428,3,1);
144:         board3D(30,431,140,452,14,1);
145:         board3D(30,456,140,473,11,1);
146:         board3D(150,409,260,428,2,1);
147:         board3D(150,431,260,452,6,1);
148:         board3D(150,456,260,473,9,1);
149:         board3D(270,431,380,452,13,1);
150:         board3D(270,456,380,473,10,1);
151:         board3D(15,250,115,320,6,1);
152:         OutThaiXY(40,274,#17'เลิกงาน');
153:         OutThaiXY(405,430,#17'โปรแกรมโดยอาจหาญ สัตยารักษ์');
154:         OutThaiXY(290,406,#17'เลือกรายการ');
155:         OutThaiXY(39,406,#17#3'ดวาดรูปวงกลม  ลบคำสั่งเดิม');
156:         OutThaiXY(39,430,#17'ลากเส้นตรง  วาดรูปสี่เหลี่ยม');
157:         OutThaiXY(39,453,#17'เครื่องคิดเลข  ป้อนข้อความ');
158:         OutThaiXY(275,430,#17'อ่านข้อสอบเดิม');
159:         OutThaiXY(275,453,#17'เก็บข้อสอบ');  mouseon;
160:        end;
161: begin
162:  ch:=#0;  Xline:=140; Yline:=60; SaveFile:=true;
163:  TestRec.bold:=1; TestRec.Cline:=14; TestRec.width:=0;
164:  NoRec:=1; TestRec.stang:=0; TestRec.Endang:=360;
165:  TestRec.Patt:=1;   TestRec.Cpatt:=12;
166:  Rx:=30; Ry:=30;    Cx:=180; Cy:=40;
167:  board3d(0,0,132,479,9,0);  showme;
168:  repeat
169:   setviewport(0,0,getmaxx,getmaxy,clipon);
170:   windowmouse(30,250,380,470);
171:   setposition(300,410);   clearmouse; Guide;
172:   repeat
173:     regs.AX:=3; Intr($33,regs); ch:=#0;
174:     if Regs.BX in [1,2] then
175:      begin
176:       Mouseoff;  findxy(xmouse,Ymouse);
177:       if (Xmouse>29) and (Xmouse<116) and
178:          (Ymouse>249) and (Ymouse<321) then Exit;
179:       if (Xmouse>29) and (Xmouse<138) then
180:         begin
181:           if (Ymouse>410) and (Ymouse<425) then ch:='C';
182:           if (Ymouse>431) and (Ymouse<449) then ch:='L';
183:           if (Ymouse>457) and (Ymouse<470) then ch:='M';
184:         end;
185:       if (Xmouse>=150) and (Xmouse<=257) then
186:         begin
187:           if (Ymouse>410) and (Ymouse<425) and
188:                                 (Norec>1) then ch:='E';
189:           if (Ymouse>431) and (Ymouse<449) then ch:='B';
190:           if (Ymouse>457) and (Ymouse<470) then ch:='W';
191:         end;
192:       if (Xmouse>270) and (Xmouse<377) then
193:         begin
194:           if (Ymouse>431) and (Ymouse<449) then ch:='R';
195:           if (Ymouse>457) and (Ymouse<470) and
196:                    (Norec>1)  and Savefile then ch:='S';
197:         end;
198:       mouseon;
199:      end;
200:   until ch in ['B','C','E','L','M','R','S','W'];
201:     Mouseoff; Mode:=Ch;
202:    case ch of
203:     'B':Editbox;
204:     'C':begin selectcircle; EditCircle; end;
205:     'E':EraseOneRec;
206:     'L':begin
207:          EditLine;
208:          Xline:=testRec.val[3]; Yline:=testRec.val[4];
209:         end;
210:     'M':begin
211:          setfillstyle(1,9); bar(10,248,117,322);
212:          calculator(6,10);
213:          windowmouse(135,5,getmaxx-16,390);
214:          setposition(300,410); showme;
215:         end;
216:     'R':begin
```

รูปที่ 7
โปรแกรมหลัก
ในการเขียนข้อสอบ

รูปที่ 7 (ต่อ)

```
217:          Mouseoff; board3D(10,400,getmaxx,getmaxy,7,0);
218:          board3D(160,410,490,470,3,1);
219:          Outthaixy(200,430,#17#3'01ที่ชื่อไฟล์ :');
220:          P:=TestRec.Cline; Testrec.Cline:=0;
221:          repeat
222:            FileName:=ReadSTR(300,430,8,'F');
223:          until filename<>'';
224:           Testrec.Cline:=P;
225:          P:=pos('.',Filename);
226:          if P<>0 then Filename:=copy(Filename,1,P)+'TST'
227:                       else Filename:=Filename+'.TST';
228:          ReadTest;
229:        end;
230:     'S':SaveTest(1);
231:     'W':Entertext;
232:    end;
233:    if Mode in ['B','L','C'] then
234:      begin
235:       TestMain[NoRec]:=TestRec; inc(NoRec);
236:       TestMain[0].val[1]:=NoRec;
237:      end;
238:  until ch=#127;
239: end;

240: {====================== EDITTEST.PAS ========================}
241: begin
242:  if Monoscn then
243:    begin closegraph; halt; end;
244:  if Pred(mouse) then
245:   begin
246:    board3D(100,120,530,230,3,0); setcolor(0);
247:    outthaixy(120,140,#17'โปรแกรมนี้ทำงานด้วยเมาส์ เขียนโดยอาจหาญ สัตยารักษ์');
248:    outthaixy(120,180,#17'   โปรดรันไฟล์ Driver ของเมาส์ก่อนการรันไฟล์นี้   ');
249:    readchar;  closegraph; halt;
250:   end;
251:  GetAspectRatio(Xasp,Yasp);  getmem(Pic,65535);
252:  Main;
253:  if (Norec>1) and SaveFile then Savetest(0);
254:  freemem(Pic,65535);
255:  freemem(ppic,imagesize(0,0,getmaxx,30));
256:  closegraph;
257: end.
```

1. ด้านซ้ายมือจะแสดงกระดาน 3 มิติ (จากบนสุดถึงล่างสุด) มีความกว้างในแนวนอน 132 จุด

2. แสดงชุดคำสั่ง Showme บริเวณมุมบนซ้าย

3. แสดงรายการเลือกของเมนูหลักที่ด้านล่าง

4. แสดงข้อความว่า เลิกงาน ไว้บนกระดาน 3 มิติ (ในข้อ 1) บริเวณกลางจอภาพ

5. ให้เม้าส์สามารถเลื่อนได้ในขอบเขตที่กำหนด เพื่อเลือกรายการต่าง ๆ ที่กำหนดบนจอภาพได้

6. ไม่มีการตรวจการกดแป้นในโปรแกรมหลักนี้

7. การเลือกรายการด้วยเม้าส์ เพื่อให้เข้าใจง่าย ผู้เขียนจะกำหนดรายการที่เลือกเป็นอักขระที่สื่อความหมายให้ทราบว่าเป็นการเลือกรายการใด เช่น C L B W ฯลฯ เป็นต้น

โปรแกรมหลักของไฟล์ edittest.pas จะเป็นดังรูปที่ 7 มีรายละเอียดดังนี้

1. บรรทัดที่ 138-160 ส่วนแสดงรายการต่าง ๆ เพื่อทำงานในเมนูหลัก โดยใช้กระดาน 3 มิติมาช่วยให้แยกแต่ละรายการออกจากกัน

2. บรรทัดที่ 162-166 กำหนดค่าให้กับตัวแปรเพื่อเริ่มต้นทำงาน

3. บรรทัดที่ 169-171 กำหนดขอบเขตของจอภาพ ขอบเขตของเม้าส์ และตำแหน่งเริ่มต้นของเม้าส์

4. บรรทัดที่ 172-200 วนรอบเลือกรายการในเมนูหลัก

5. บรรทัดที่ 177-178 กดปุ่มของเม้าส์บริเวณข้อความ เลิกงาน

6. บรรทัดที่ 181 กดปุ่มของเม้าส์บริเวณข้อความ วาดรูปวงกลม

7. บรรทัดที่ 182 กดปุ่มของเม้าส์บริเวณข้อความ ลากเส้นตรง

8. บรรทัดที่ 183 กดปุ่มของเม้าส์บริเวณข้อความ เครื่องคิดเลข

9. บรรทัดที่ 187-188 กดปุ่มของเม้าส์บริเวณข้อความ ลบคำสั่งเดิม

10. บรรทัดที่ 189 กดปุ่มของเม้าส์บริเวณข้อความ วาดรูปสี่เหลี่ยม

11. บรรทัดที่ 190 กดปุ่มของเม้าส์บริเวณข้อความ ป้อนข้อความ

12. บรรทัดที่ 194 กดปุ่มของเม้าส์บริเวณข้อความ อ่านข้อสอบเดิม

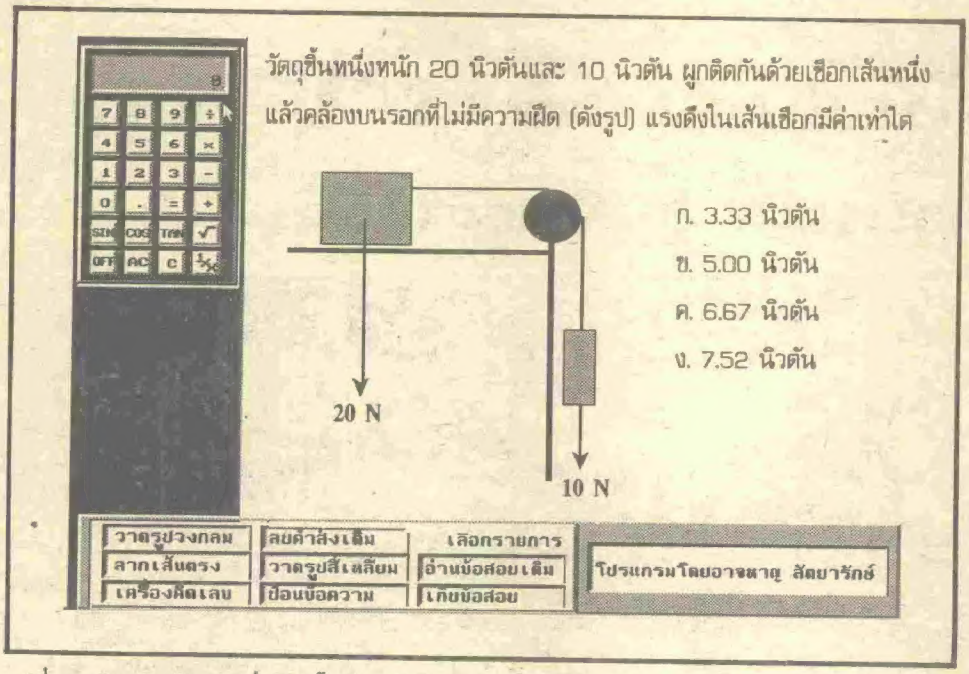
13. บรรทัดที่ 195-196 กดปุ่มของเม้าส์บริเวณข้อความ เก็บข้อสอบ

14. บรรทัดที่ 203-231 การทำงานเมื่อเลือกรายการต่าง ๆ

15. บรรทัดที่ 233-237 ในกรณีที่เป็นการวาดรูปสี่เหลี่ยม ลากเส้นตรง วาดวงกลม จะทำการเก็บข้อมูลลงตัวแปร TestMain พร้อมกับเพิ่มค่าตัวแปรที่เก็บจำนวนระเบียนข้อมูล

16. บรรทัดที่ 241-257 ส่วนการกำหนดเงื่อนไขในการทำงาน และเรียกใช้โปรแกรมหลักให้ทำงาน

ผู้เขียนได้ทดลองเขียนข้อสอบ 1 ข้อ โดยลอกหน้าจอภาพมาให้ดูเป็นตัวอย่างด้วยคือรูปที่ 8 ซึ่งได้ใช้คำสั่งทั้งหมดที่มีให้ เพื่อให้เป็นโปรแกรมที่สมบูรณ์จึงได้เลือกใช้เครื่องคิดเลขด้วย ผู้อ่านจึงสามารถเห็นส่วนที่ปรากฏในเมนูหลักได้ทุกรายการ รวมรูปร่างของเครื่องคิดเลขด้วย



รูปที่ 8 ตัวอย่างข้อสอบที่สร้างขึ้นด้วยไฟล์ edittest.exe

การสร้างข้อสอบนี้ไม่ต้องป้อนเลขของข้อสอบ เพราะเราจะนำข้อสอบข้อใดมาเป็นข้อ 1 หรือ 2 หรือ 3 ก็ได้ โดยหมายเลขข้อสอบนี้ผู้เขียนจะให้ไฟล์สำหรับผู้เข้าสอบ (ไฟล์ test.exe) แสดงลำดับเองทั้งหมด

ก่อนจาก

ระหว่างที่รอวารสารไมโครฯฉบับต่อไปวางตลาด ตอนที่ 3 ของบทความขอให้ผู้อ่านทำการสร้างข้อสอบเก็บไว้อย่างน้อย 10 ข้อ เพื่อนำไปใช้ทดสอบไฟล์ test.pas ในบทความตอนต่อไป หรือผู้อ่านจะลองเขียนขึ้นมาเองก่อนก็ได้ โดยแนวที่ผู้เขียนกำหนดไว้ในการสอบมีดังนี้

1. เขียนไฟล์เพื่อเก็บรายการของข้อสอบรวม ให้มีส่วนสำคัญคือ

- เวลาการสอบทั้งฉบับ (ไม่เกิน 120 นาที)
- จำนวนข้อสอบทั้งหมด
- ชื่อหลักของข้อสอบชุดนั้น (ไม่เกิน 5 ไบต์) เพราะเราเอาเลข 1-100 มาแทรกต่อท้ายชื่อไฟล์ในตอนรันโปรแกรม

การที่กำหนดชื่อหลักของข้อสอบนั้น จะสะดวกกว่ากำหนดชื่อไฟล์ข้อสอบ (เพราะไม่ต้องกำหนดทุกไฟล์) รวมทั้งการอ่านไฟล์ข้อสอบทั้งชุดมาให้ผู้เข้าสอบก็ง่าย

ไฟล์ที่เก็บข้อมูลนี้จะให้ชื่อ maintest.dta

2. ไฟล์ test.pas จะทำงานดังนี้

- อ่านข้อมูลจากไฟล์ maintest.dta เข้ามาก่อน
- ทำการสลับลำดับที่ของข้อสอบ (เพื่อให้ผู้สอบแต่ละคนเสมือนได้ข้อสอบคนละฉบับ)
- เก็บเวลาเริ่มต้นการทำข้อสอบ เก็บวันเดือนปีของการสอบ ป้อนชื่อผู้สอบ
- ผู้เข้าสอบสามารถข้ามการทำข้อสอบข้อที่ยังคิดไม่ออกไปได้
- ผู้เข้าสอบสามารถย้อนมาทำข้อสอบเก่าได้ ไม่ว่าจะเคยทำไปแล้วหรือยังไม่ได้ทำ
- เมื่อผู้เข้าสอบสิ้นสุดการสอบของตนเอง และเลือกรายการ ส่งข้อสอบ โปรแกรมจะทำการสลับลำดับที่ข้อสอบคืนเป็นลำดับดังเดิม (เพื่อให้ข้อมูลที่เก็บเป็นข้อสอบชุดเดียวกัน) เก็บเวลาสิ้นสุดการสอบ ตรวจให้คะแนน และเขียนข้อมูลลงในไฟล์ test.dta

3. เขียนไฟล์ readtest.pas เพื่ออ่านข้อมูลจากไฟล์ test.dta เพื่อดูผลการสอบ

ผู้เขียนหวังว่า ผู้อ่านคงได้ทดลองเขียนโปรแกรมในการสอบขึ้นมาใช้เองก่อน เพราะผู้อ่านอาจเขียนได้ดีกว่าที่จะเสนอในบทความตอนที่ 3 ก็ได้ ซึ่งจะช่วยให้ท่านมีความมั่นใจในการพัฒนาโปรแกรมคอมพิวเตอร์ด้วยตนเองมากยิ่งขึ้น

สำหรับฉบับนี้คงต้องบอกว่า โชคดีครับ...